# WEBADM SAML IDENTITY PROVIDER

# WebADM SAML Identity Provider

Web-Service SSO Federation

## Configuration of WebADM as a SAML Identity Provider

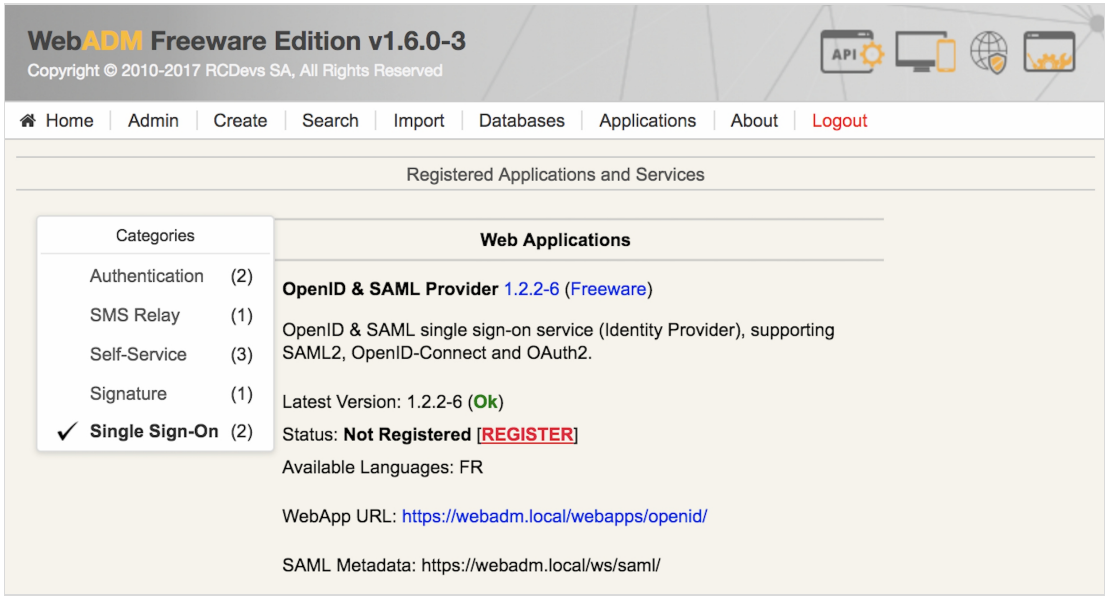
### 1. Configuration of the Identity Provider

First, we need a WebADM server with *MFA Authentication Server* and *OpenID & SAML Provider*. We can use the appliance or install a new server.

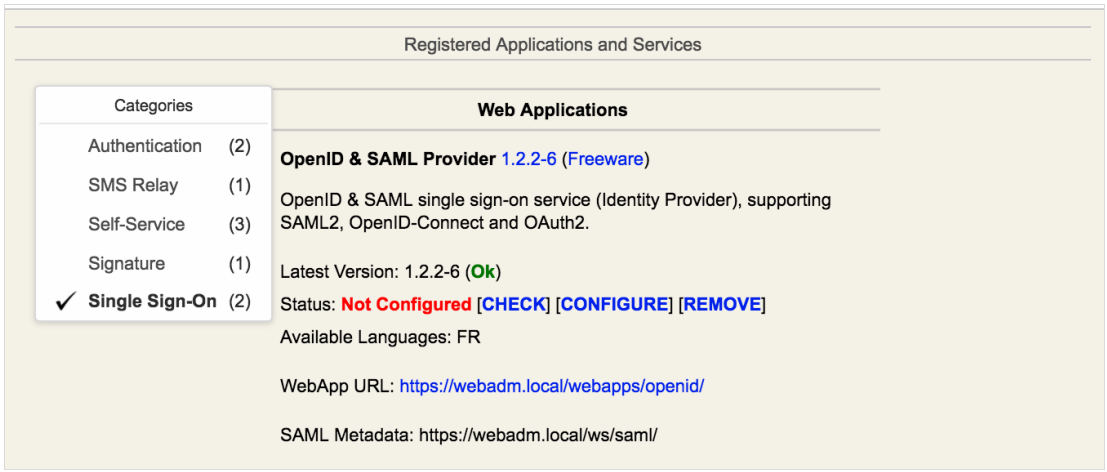
We need also a DNS name for the server. If we can not change the DNS, we can also add the name in `/etc/hosts` or `c:\WINDOWS\system32\drivers\etc\hosts` for testing purpose:

Once the server is up and running, we can configure it as a SAML Identity Provider (IdP).

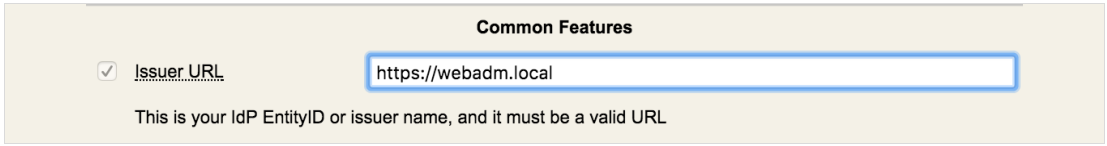
We connect to the WebADM GUI > `Applications` tab > `Singe Sign-On` > `OpenID & SAML Provider` > `REGISTER`:



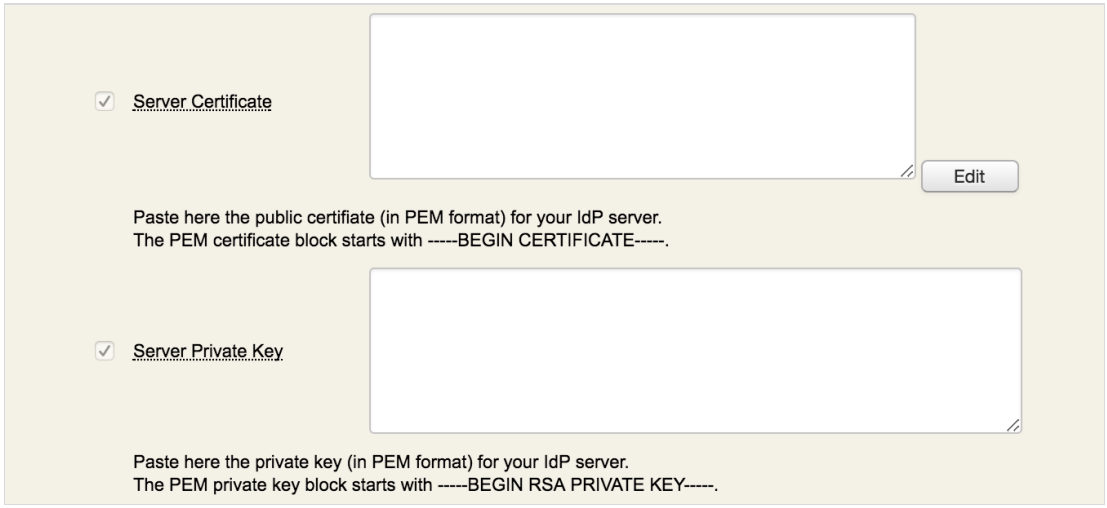

We click on `CONFIGURE`:



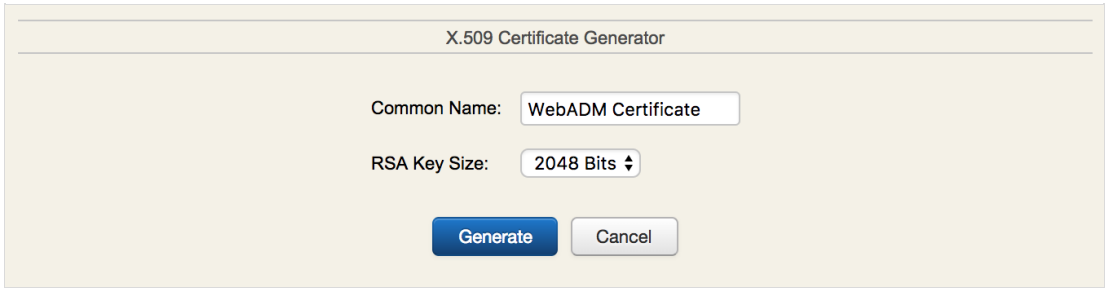

We add the url of the server in `Issuer URL`:



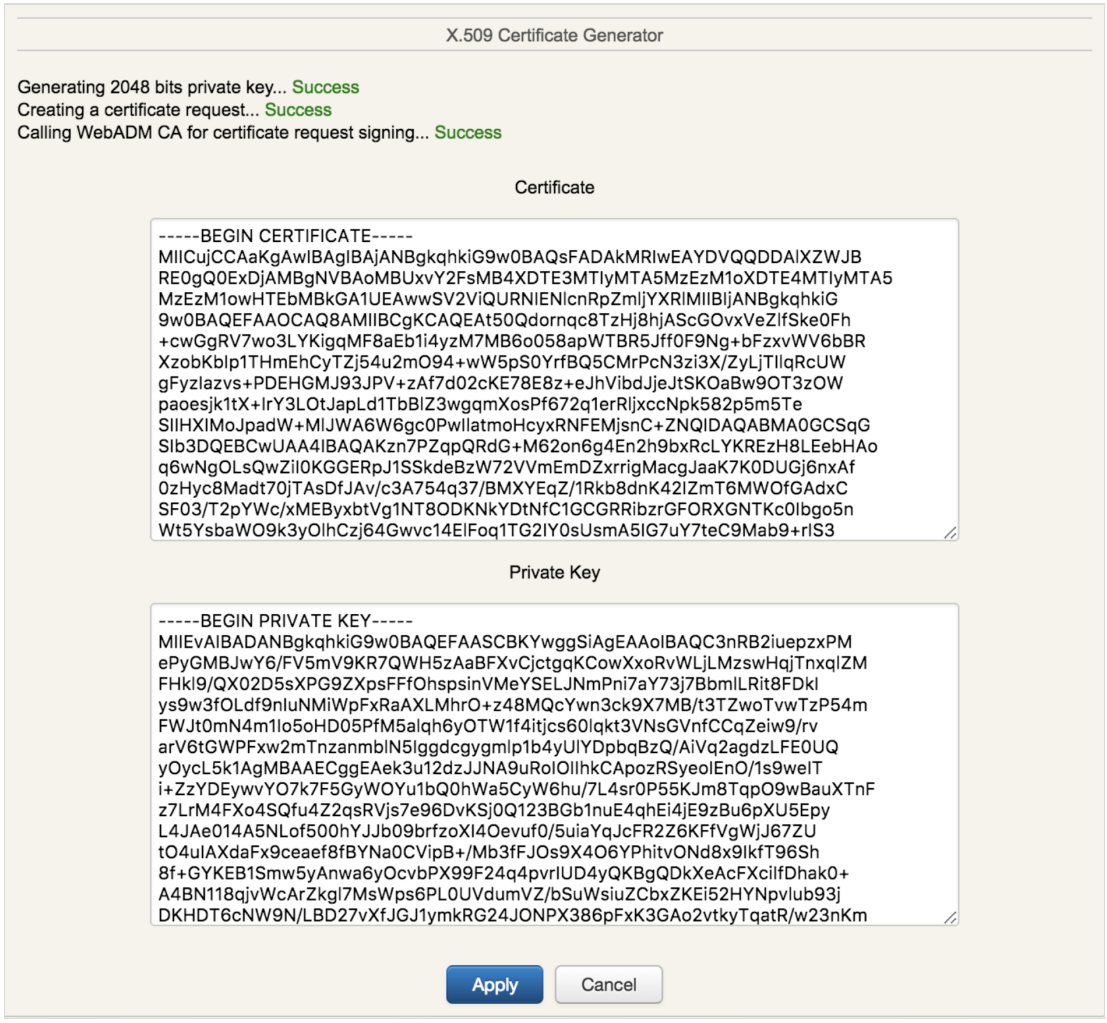

At `Server Certificate`, we click on `Edit`:



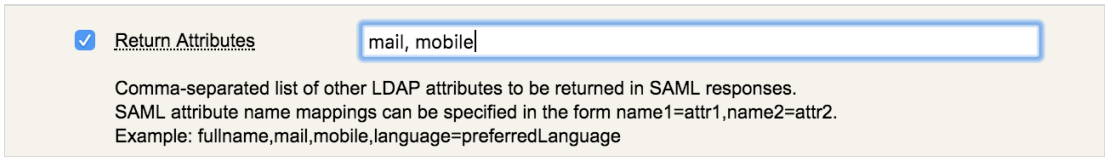

We click on `Generate`:



Now, we have the IdP certificate, we click on `Apply`:




We can add some extra attributes, for example `mail` and `mobile`, and click on `Apply`:




That's all, WebADM is now able to work as an Identity Provider (IdP).

We can check metadata, go to WebADM GUI > `Applications` > `Single Sign-On` > `OpenID & SAML Provider` > `SAML Metadata` and open the link in a new tab:

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="https://webadm.local">
<IdPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<KeyDescriptor use="signing">
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<X509Data>
<X509Certificate>
MIICujCCAaKgAwIBAgIBAzANBgkqhkiG9w0BAQsFADAkMRIwEAYDVQQDDAlXZWJBRE0gQ0ExDjAMBgNVBAoMBUxv\

</X509Certificate>
<!--
 Cert Fingerprint (SHA1): 802b0a629dfc11a686306a73f8b11b272e1b9ca2
-->
<!--
 Cert Fingerprint (MD5): a0480b3a54a7ea7e2da2d6b9e27fbfbf
-->
</X509Data>
</KeyInfo>
</KeyDescriptor>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://webadm.local/webapps/openid/"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://webadm.local/webapps/openid/"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://webadm.local/webapps/openid/"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://webadm.local/webapps/openid/"/>
</IdPSSODescriptor>
</EntityDescriptor>
```

## 2. Configuration of a Service Provider: SP Initiated

For this test, we are using simplesamplphp.

We install it on another *CentOS 7* server.

We open http port:

```
firewall-cmd --permanent --add-service http
firewall-cmd --reload
```

We disable selinux:

```
setenforce 0
vi /etc/selinux/config
```

We install required packages:

```
yum install wget php php-mbstring php-xml httpd
```

We install *simplesamlphp*:

```
wget "https://simplesamlphp.org/download?latest" -O ssp.tgz
tar xzf ssp.tgz
mv simplesamlphp* /var/simplesamlphp
```

We add a virtual host to *Apache* (replace *sp.local* with the right DNS name who point to this server):

```
vi /etc/httpd/conf.d/saml.conf
```

```
<VirtualHost *>
        ServerName sp.local
        DocumentRoot /var/www/sp.local

        SetEnv SIMPLESAMLPHP_CONFIG_DIR /var/simplesamlphp/config

        Alias /simplesaml /var/simplesamlphp/www

        <Directory /var/simplesamlphp/www>
            Require all granted
        </Directory>
</VirtualHost>
```

We add the Identity Provider. All these values should correspond to the content of metadata from SAML configuration in WebADM:

- *$metadata[ ]* corresponds to *entityID*
- *SingleSignOnService* corresponds to *SingleSignOnService Location=*
- *SingleLogoutService* corresponds to *SingleLogoutService Location=*
- *certFingerprint* corresponds to *Cert Fingerprint (SHA1)*

```
vi /var/simplesamlphp/metadata/saml20-IdP-remote.php
```

```
<?php
  $metadata['https://webadm.local'] = array(
    'SingleSignOnService'  => 'https://webadm.local/webapps/openid/',
    'SingleLogoutService'  => 'https://webadm.local/webapps/openid/',
    'certFingerprint'      => '802b0a629dfc11a686306a73f8b11b272e1b9ca2',
);
```

We enable *SAML* in `/var/simplesamlphp/config/config.php`:

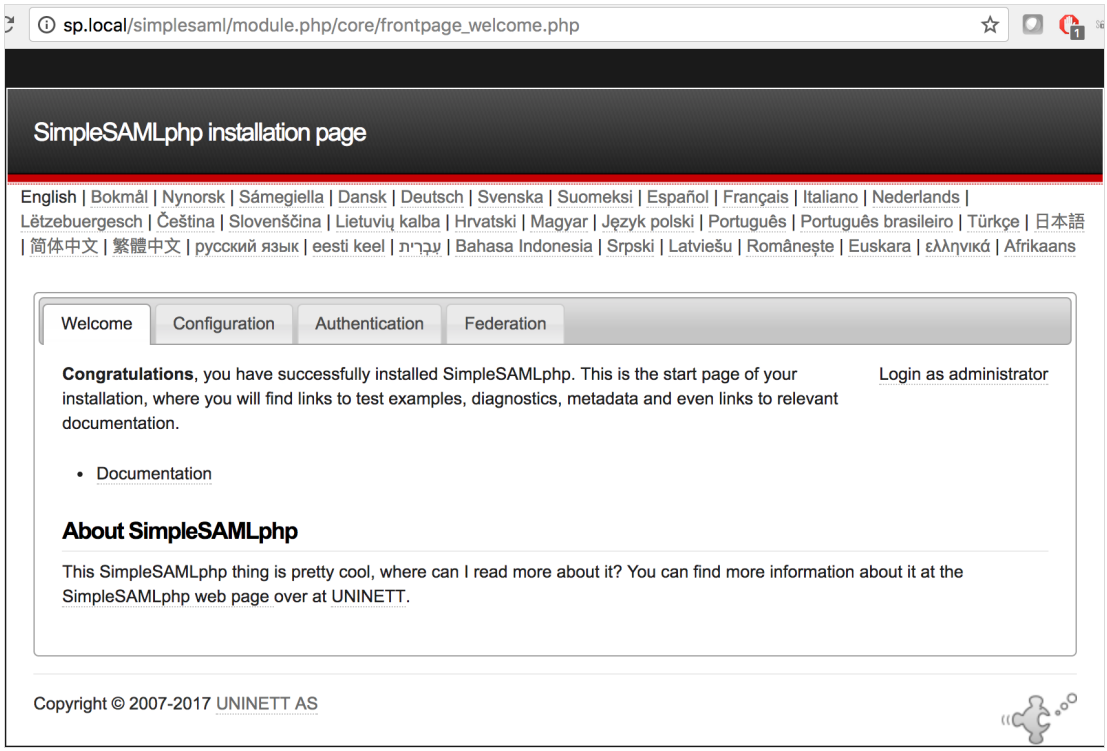
```
vi /var/simplesamlphp/config/config.php
```

```
enable.saml20-IdP' => true
```

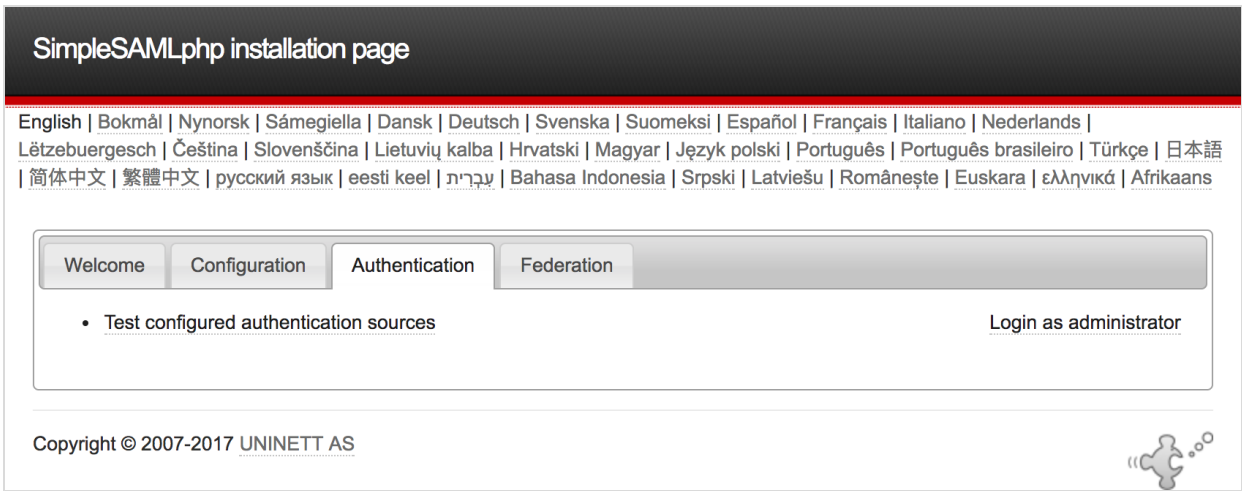
We start *Apache*:

```
systemctl start httpd
systemctl enable httpd
```

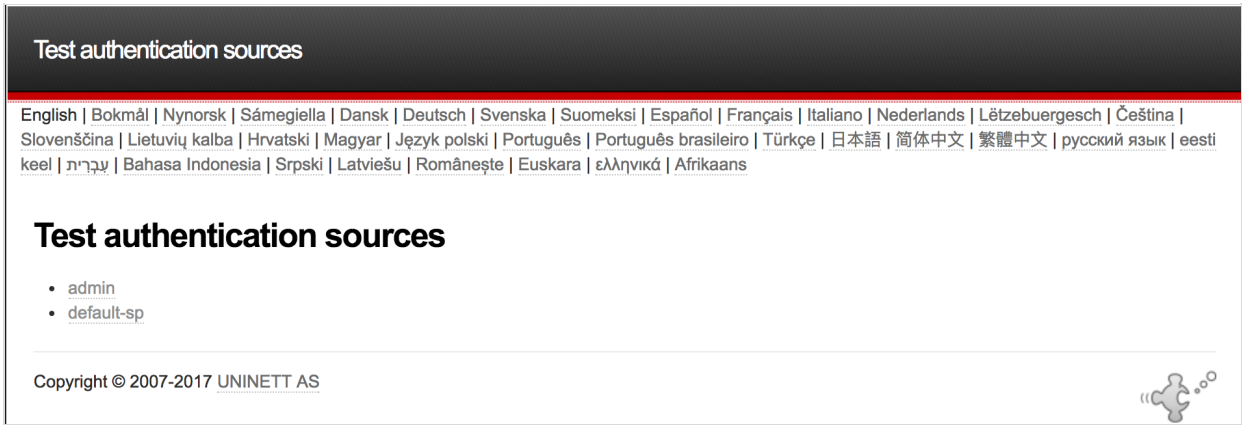
We open `http://sp.local/simplesaml` in a browser:



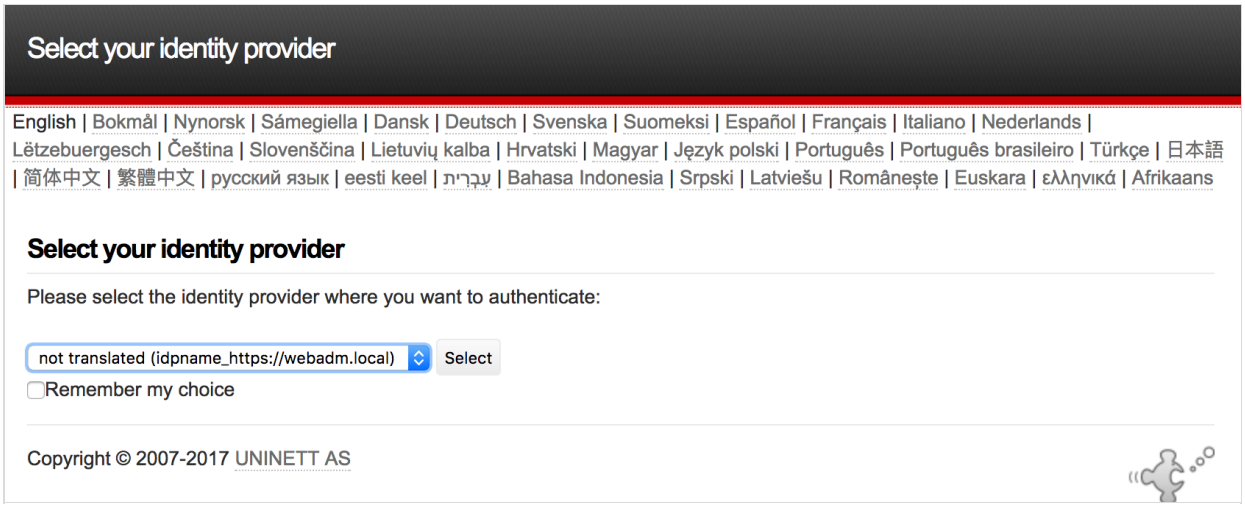

We click on `Authentication`:




We click on `Test configured authentication sources`:

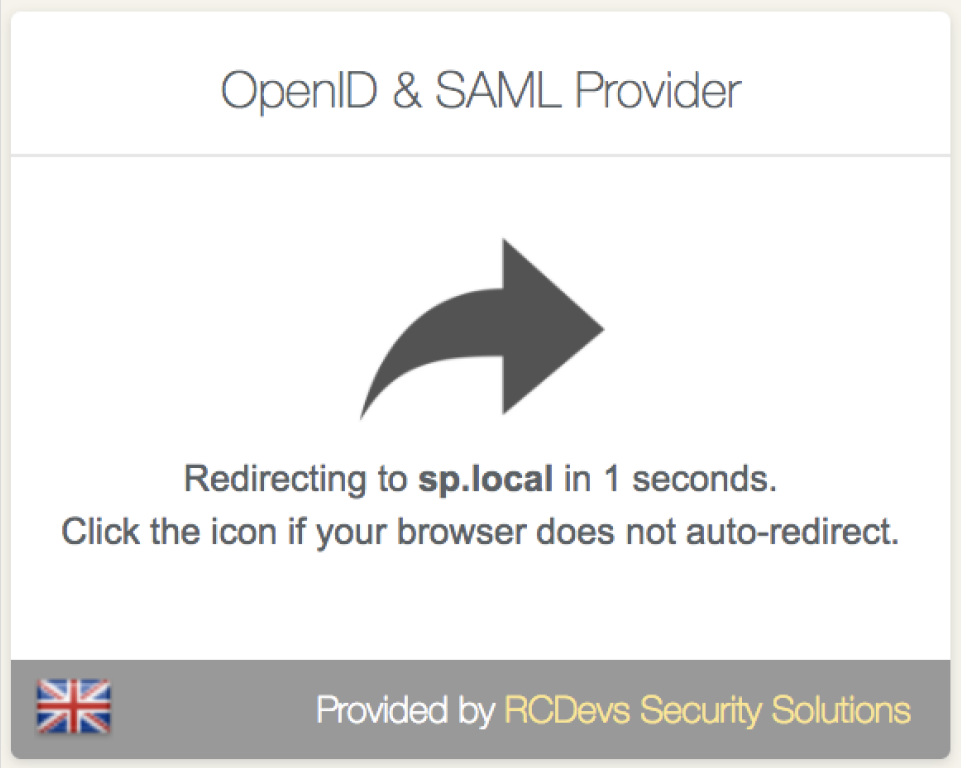



We click on `default-sp`:

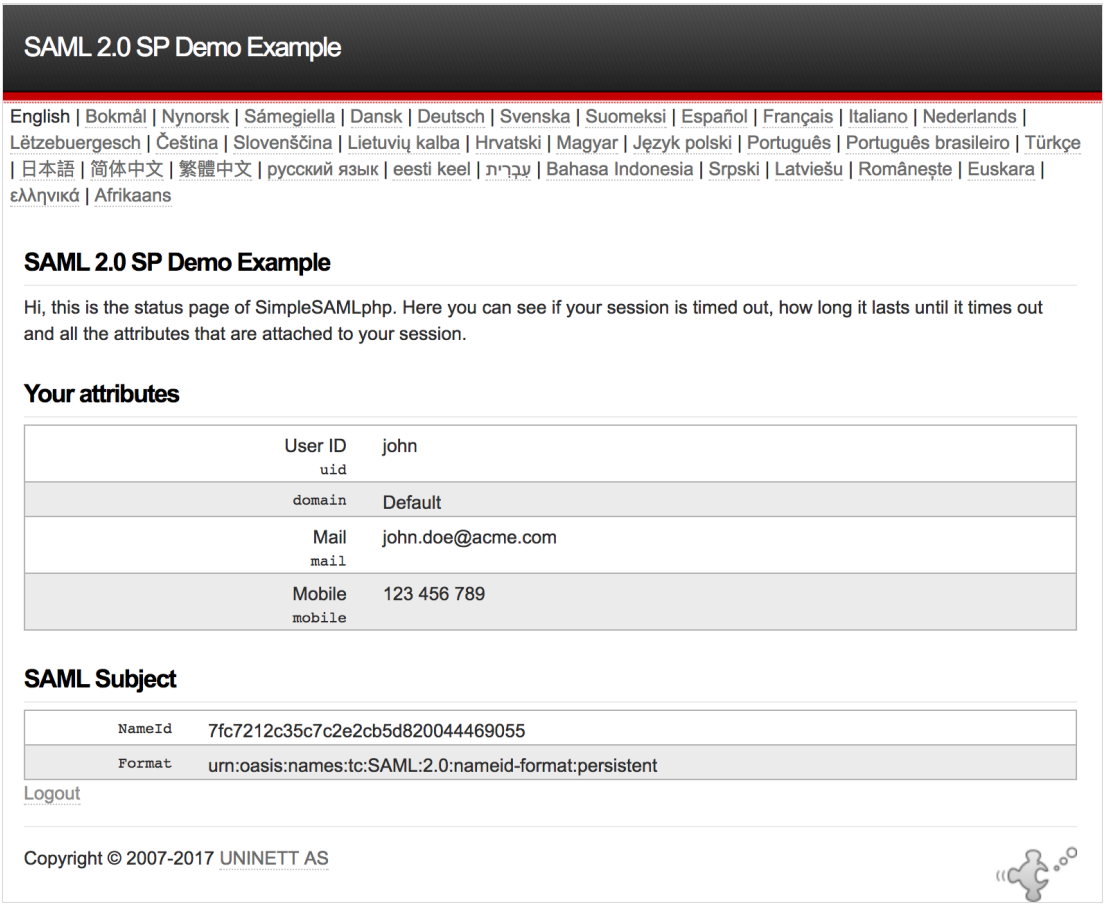



We click on `Select`:


We authenticate with an activated user through WebADM IdP:




It's done, we are authenticated:




We can check the log in `/opt/webadm/logs/webadm.log`:

```
[2017-12-21 11:16:31] [192.168.1.220] [OpenID:Y84I9XHY] User not authenticated
(entering login form)
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] New login request (OpenOTP)
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] > Username: john
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] > Domain: Default
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] > ANY Password: xxxxxxx
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] Sending openotpSimpleLogin
request
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] New openotpSimpleLogin SOAP
request
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Username: john
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Domain: Default
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Password: xxxxxxx
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Client ID: OpenID
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Source IP: 192.168.1.220
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Context ID:
5cf415099b146265083580f7098f5717
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Registered openotpSimpleLogin
request
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Resolved LDAP user: cn=john,o=Root
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Started transaction lock for user
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 1 user mobiles: 123 456 789
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 1 user emails:
john.doe@acme.com
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 37 user settings:
LoginMode=LDAP,OTPType=TOKEN,OTPLength=6,ChallengeMode=Yes,ChallengeTimeout=90,MobileTime
1:HOTP-SHA1-6:QN06-
T1M,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,LastOTPTime=300,ListChallengeMode=S

[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 1 user data: LoginCount
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Requested login factors: LDAP
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] LDAP password Ok
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Updated user data
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Sent success response
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] OpenOTP authentication success
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] Resolved LDAP user:
cn=john,o=Root (cached)
[2017-12-21 11:16:37] [192.168.1.220] [OpenID:7TWF4J4E] Login session started for
cn=john,o=Root
[2017-12-21 11:16:37] [192.168.1.220] [OpenID:7TWF4J4E] Sent SAML success response
```

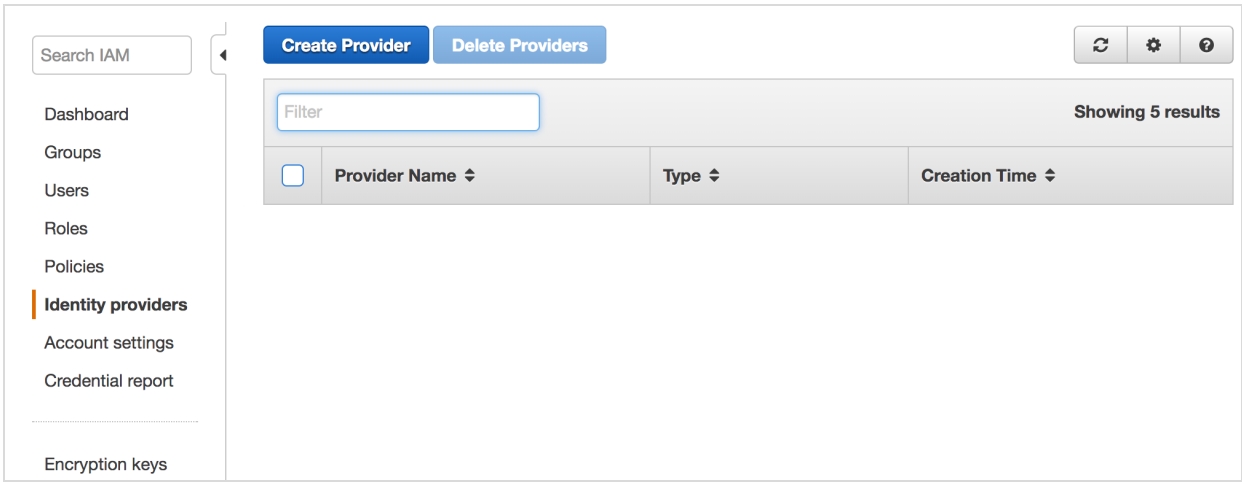
## 3. Configuration of a Service Provider: IdP Initiated

In this case, the authentication will be started directly from *OpenID & SAML Provider* web application. We will configure WebADM to manage authentications with Amazon Web Service (AWS). Other Service providers are available but not shown in this HowTo: GSuite, SalesForce, SugarCRM, Zimbra, GoToMeeting, GoToWebinar, GoToTraining and GoToAssist.

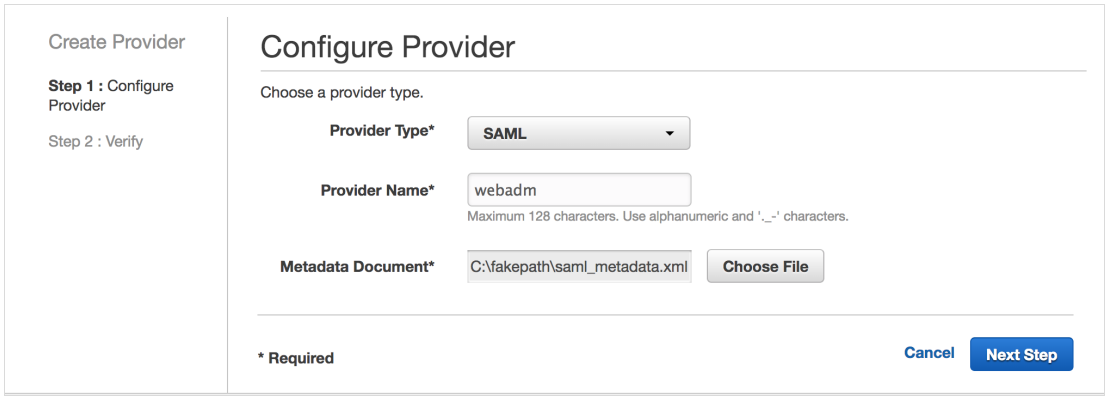
### 3.1 Configuration of AWS

First, we save the SAML metadata in a file. For our IdP server, we find it in `https://webadm.local/ws/saml/`.

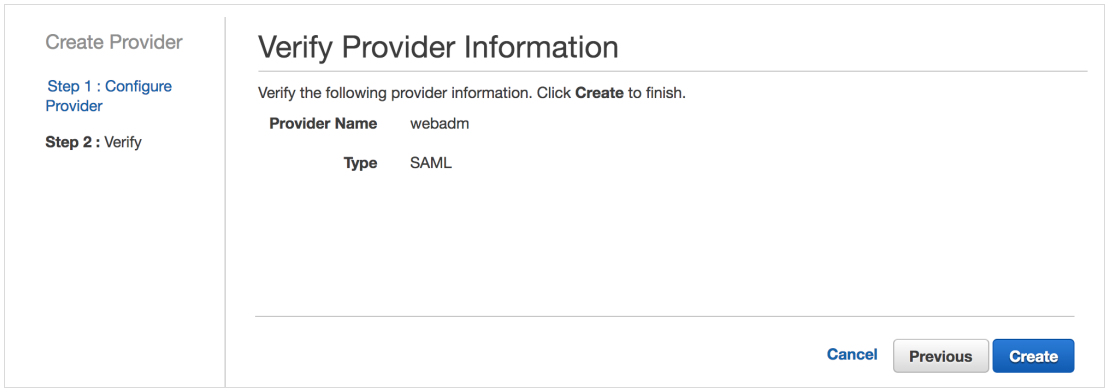
We open AWS console > `IAM` > `Identity providers` > `Create Provider`:



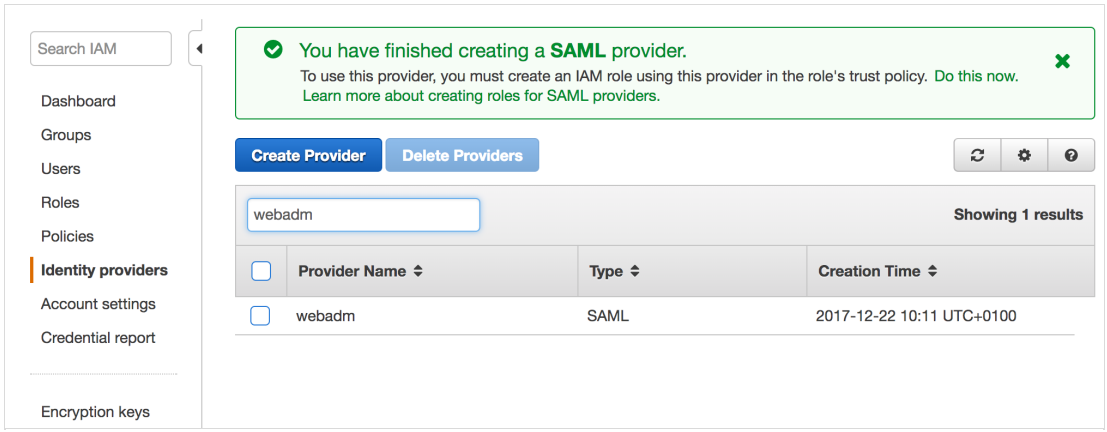

We select `SAML`, add a name, insert the metadata file and click on `Next Step`:



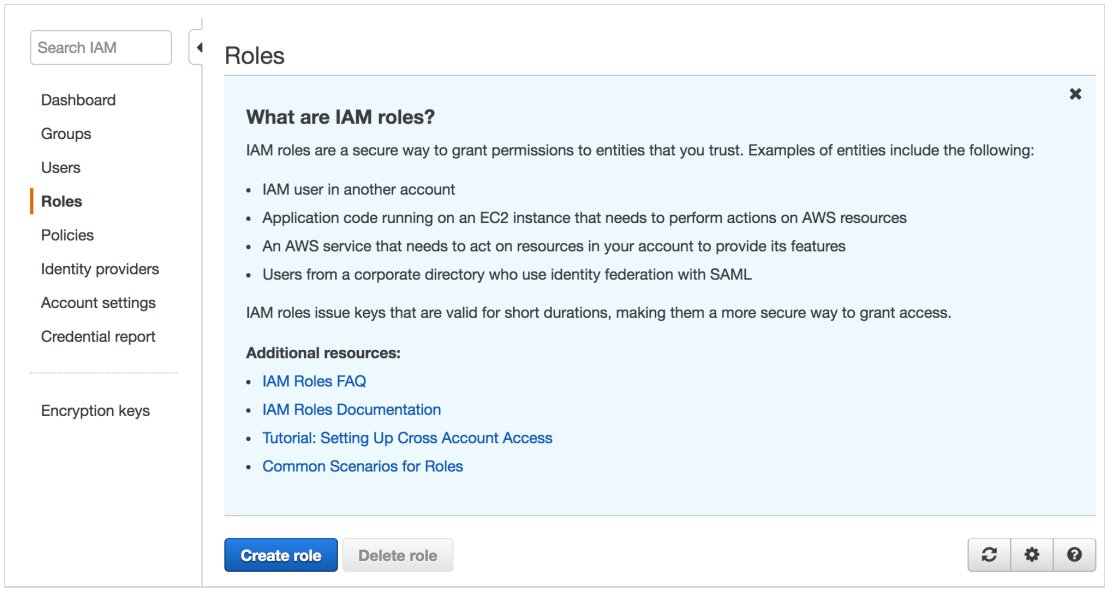

We click on `Create`:



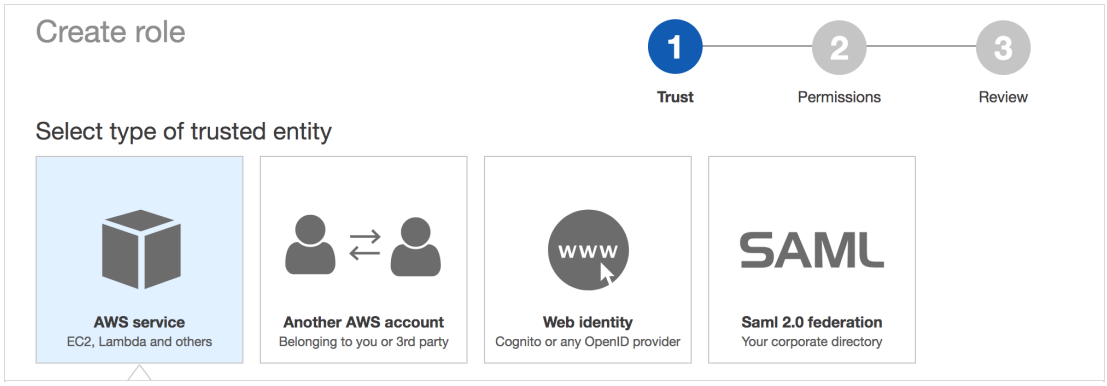

Now, our IdP is added to AWS. We select `Roles`:



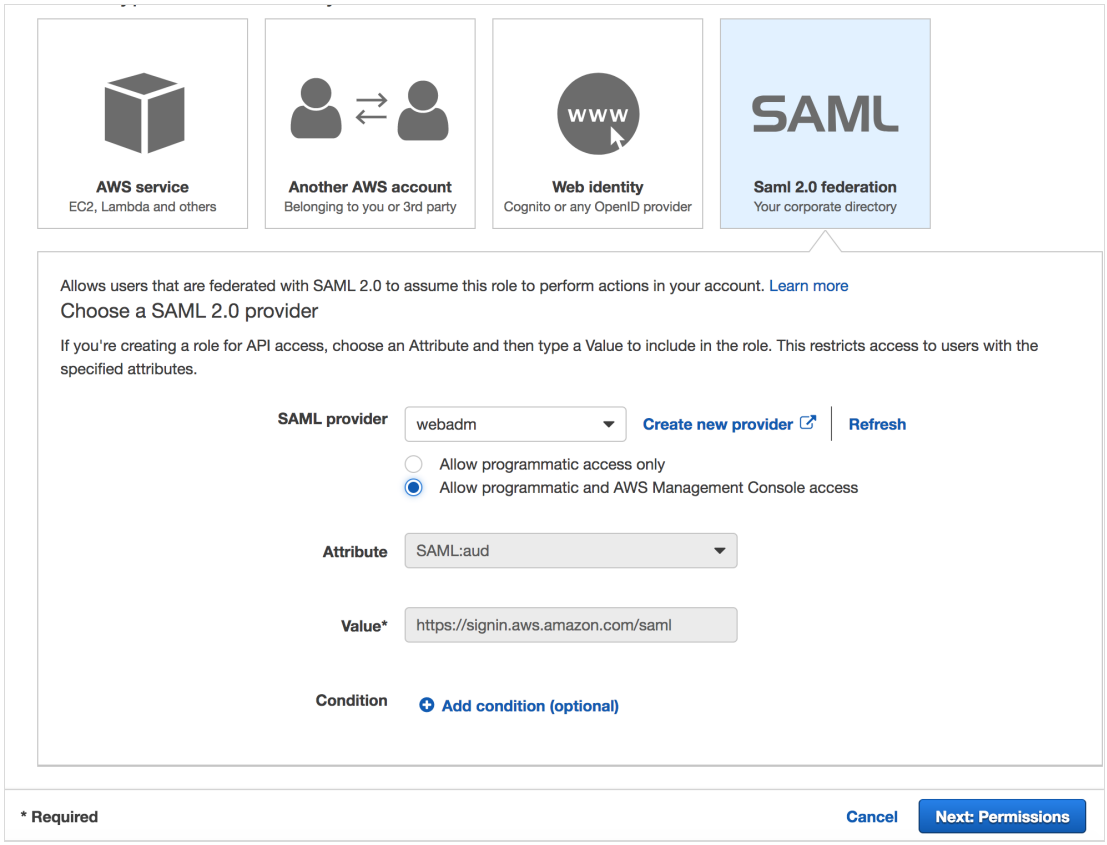

We click on `Create Role`:



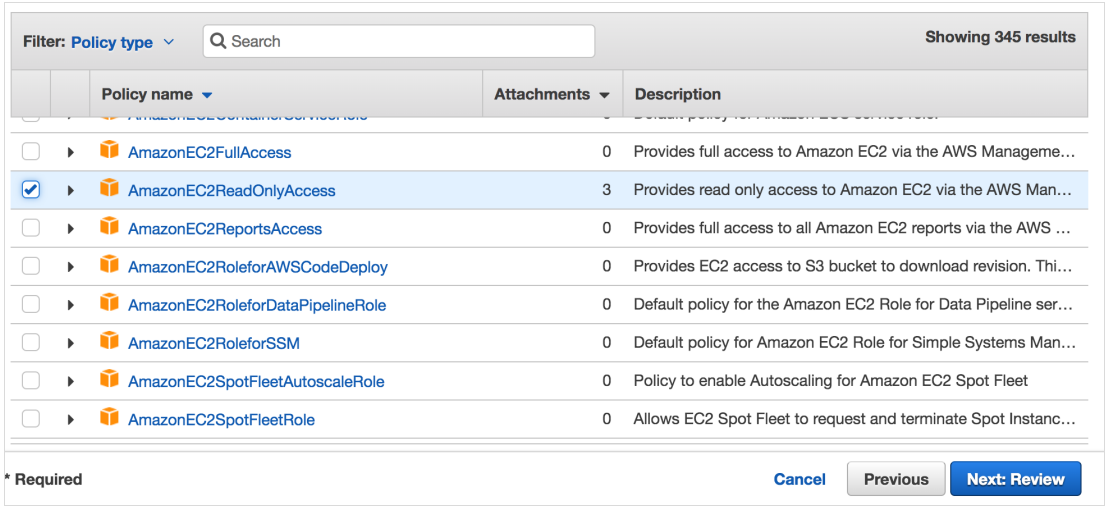

We click on `SAML`:



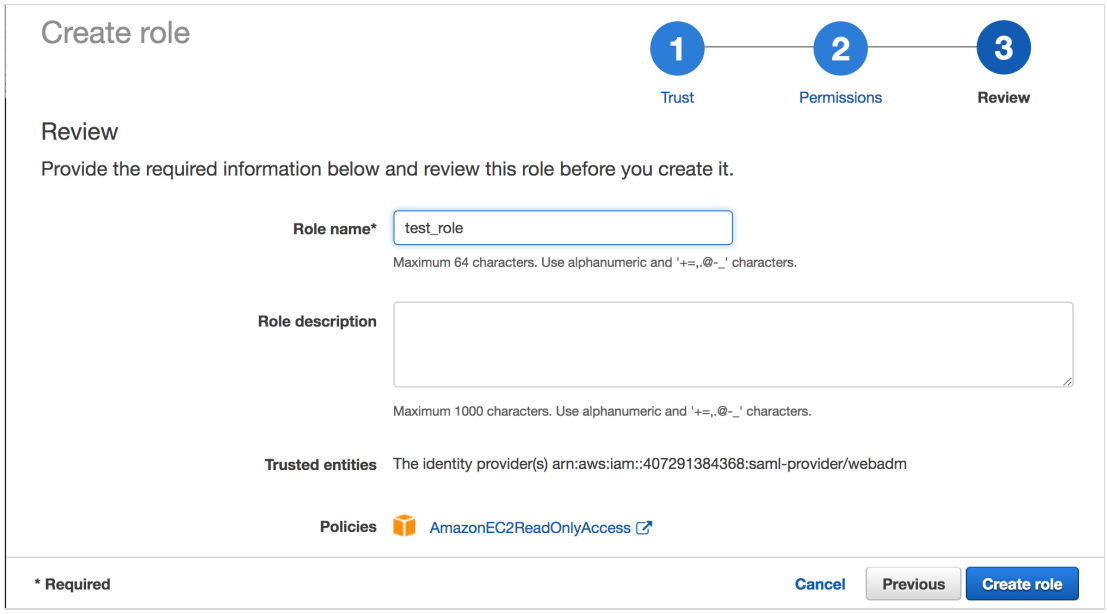

We select our SAML provider, select `AWS Management Console access` and click on `Next Permission`:



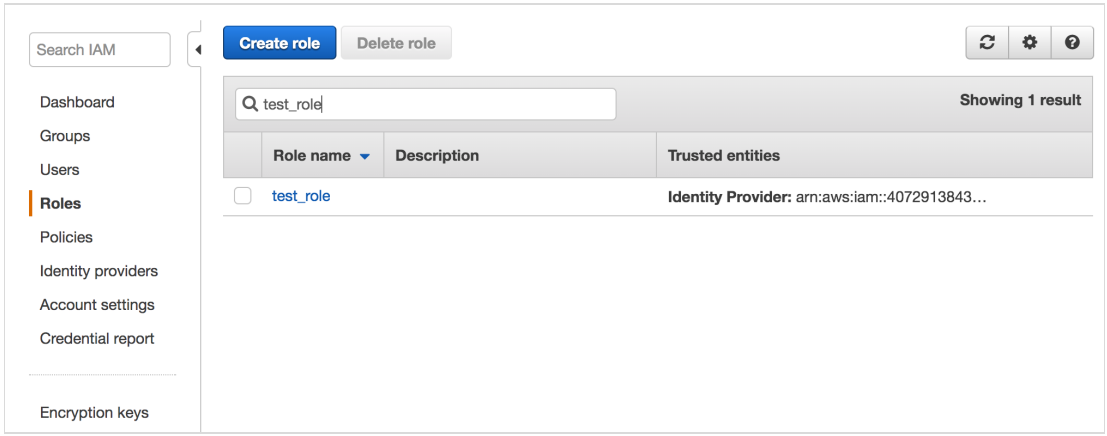

We select a permission policy and click on `Next: Review`.

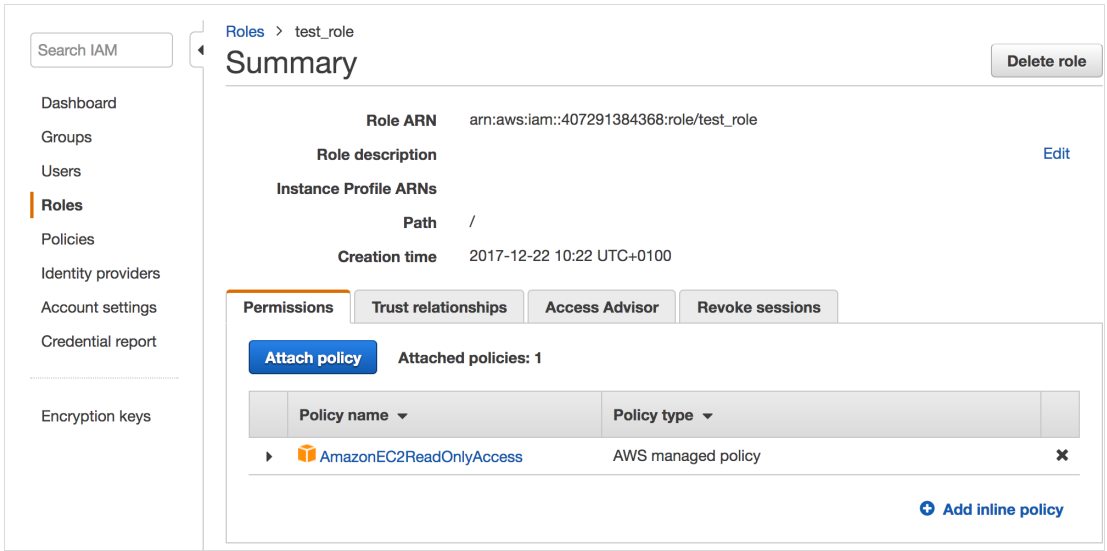



We add a name and click on `Create role`:




The role is now created, we can select it to see more details.

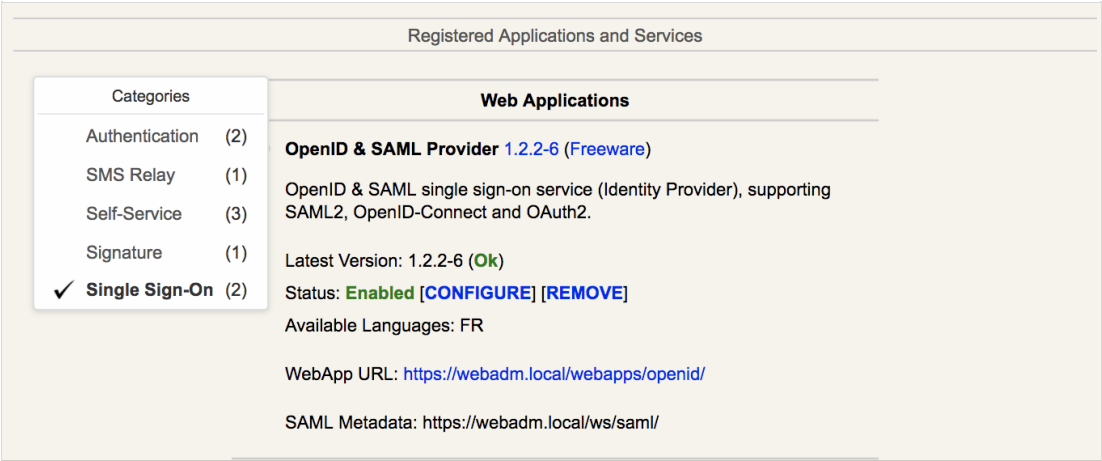


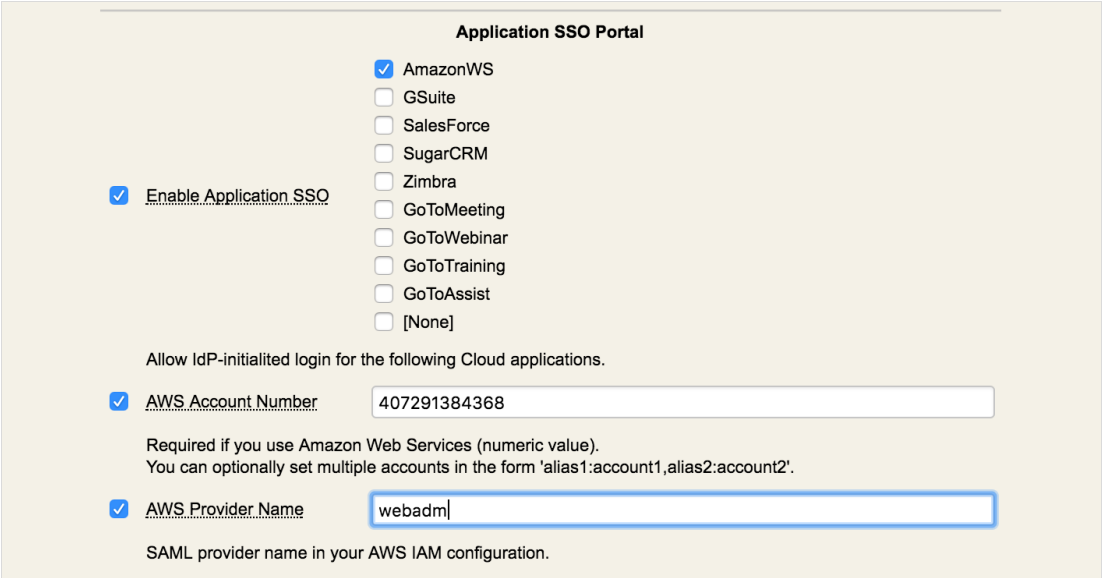

## 3.2 Configuration of WebADM

We need to activate IdP initiated authentication for AWS.

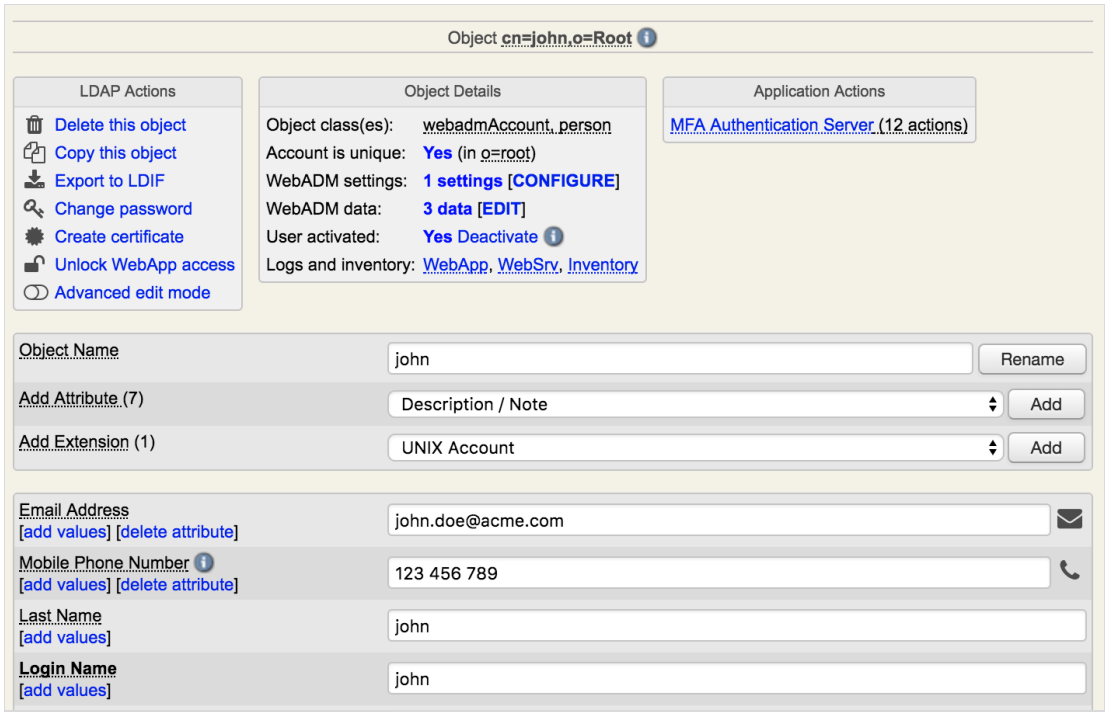
We open the configuration in WebADM GUI > `Applications` > `Single Sign-on` > `CONFIGURE`:



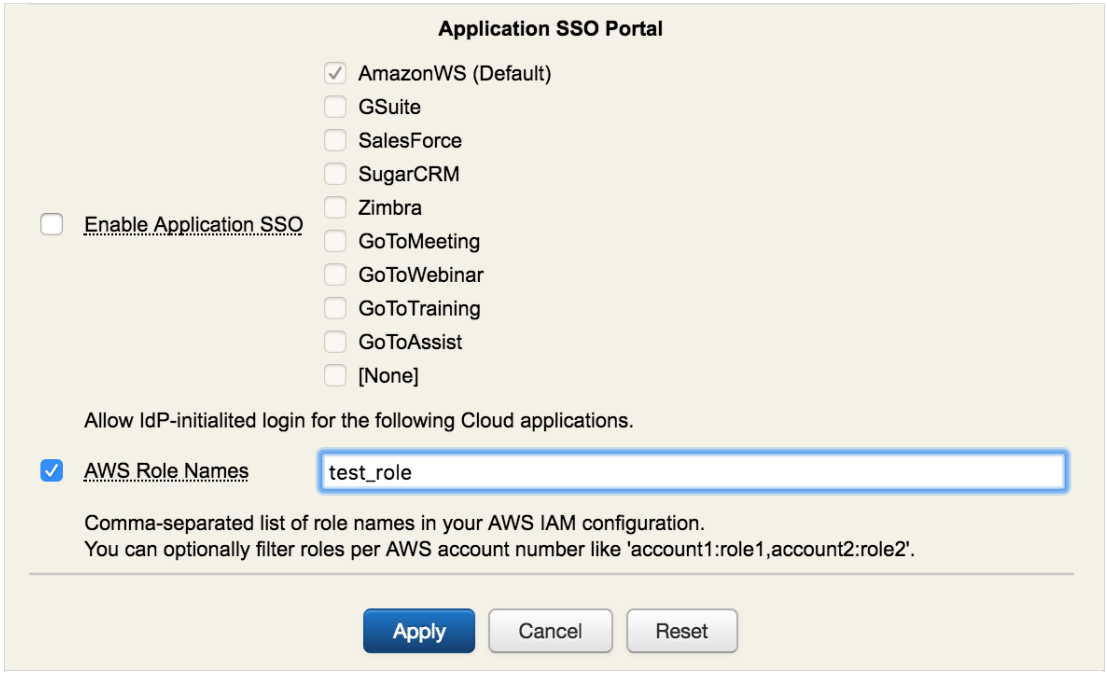

We check `Enable Application SSO` and `AmazonWS`, we add `AWS Account Number` (a numerical value that you can find in the ARN of the AWS role) and `AWS Provider Name` and apply:




We select the test user and click on `WebADM settings: [CONFIGURE]`:

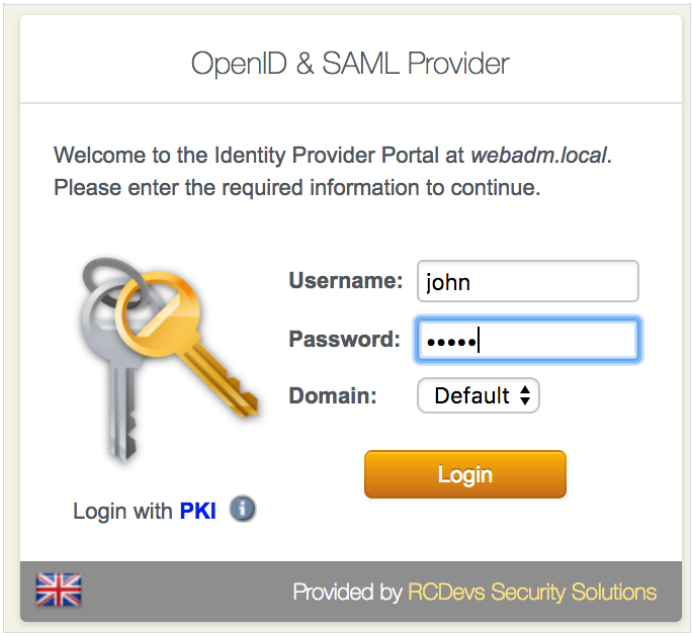



We select `OpenID`, add `AWS Role Names` and `Apply`. We can also add the AWS role to an LDAP group:

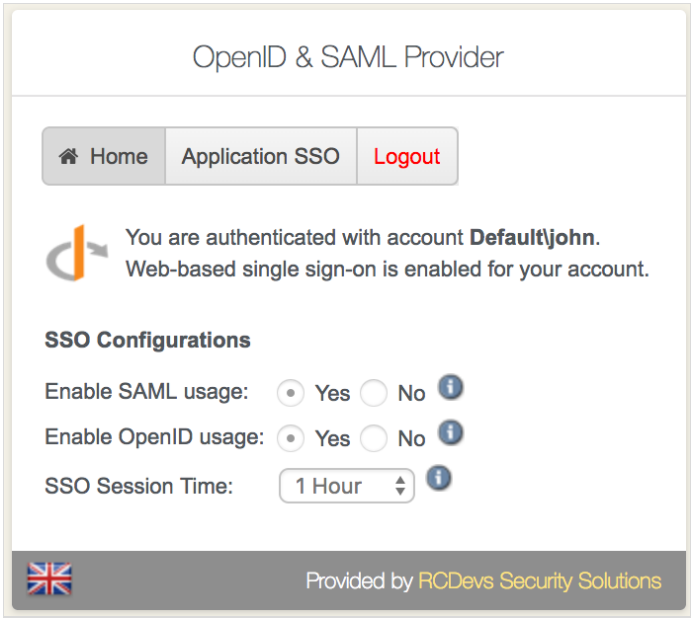



## 3.3 Testing

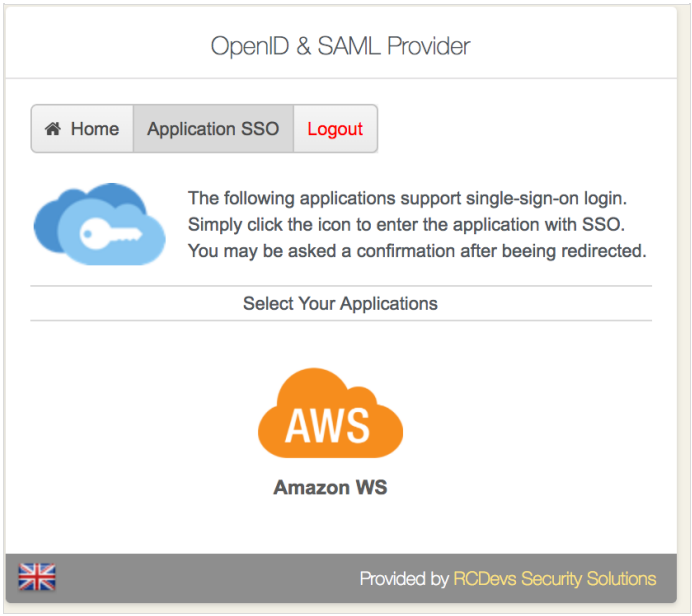
We open the web application in `https://webadm.local/webapps/openid/` and `Login` with the user:

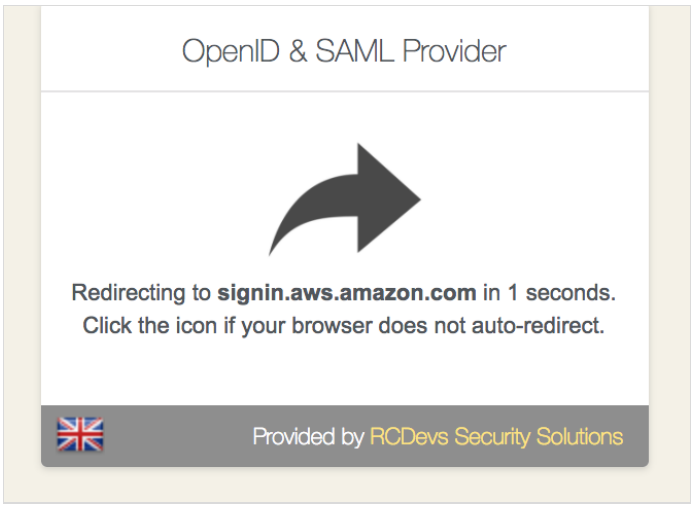



We select `Application SSO`:

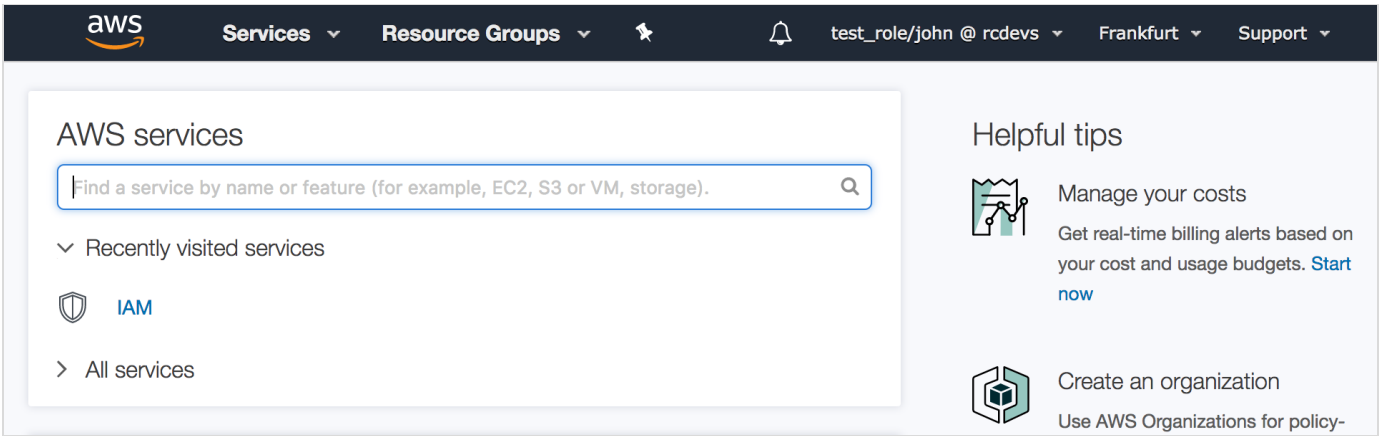



We click on `Amazon WS`:




That's it,we are now connected to AWS:




We can check the log in `/opt/webadm/logs/webadm.log`:

```
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] New login request (OpenOTP)
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] > Username: john
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] > Domain: Default
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] > ANY Password: xxxxxxx
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] Sending openotpSimpleLogin
request
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] New openotpSimpleLogin SOAP
request
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Username: john
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Domain: Default
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Password: xxxxxxx
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Client ID: OpenID
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Source IP: 192.168.1.220
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Context ID:
5cf415099b146265083580f7098f5717
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] Registered openotpSimpleLogin
request
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] Resolved LDAP user: cn=john,o=Root
(cached)
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Started transaction lock for user
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 1 user mobiles: 123 456 789
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 1 user emails:
john.doe@acme.com
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 37 user settings:
LoginMode=LDAP,OTPType=TOKEN,OTPLength=6,ChallengeMode=Yes,ChallengeTimeout=90,MobileTime
1:HOTP-SHA1-6:QN06-
T1M,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,LastOTPTime=300,ListChallengeMode=S

[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 2 user data:
LoginCount,RejectCount
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Requested login factors: LDAP
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] LDAP password Ok
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Updated user data
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Sent success response
[2017-12-22 09:35:18] [192.168.1.220] [OpenID:4JGOGC0T] OpenOTP authentication success
[2017-12-22 09:35:18] [192.168.1.220] [OpenID:4JGOGC0T] Resolved LDAP user:
cn=john,o=Root (cached)
[2017-12-22 09:35:18] [192.168.1.220] [OpenID:4JGOGC0T] Login session started for
cn=john,o=Root
[2017-12-22 09:36:50] [192.168.1.220] [OpenID:4JGOGC0T] Sent SAML success response
```

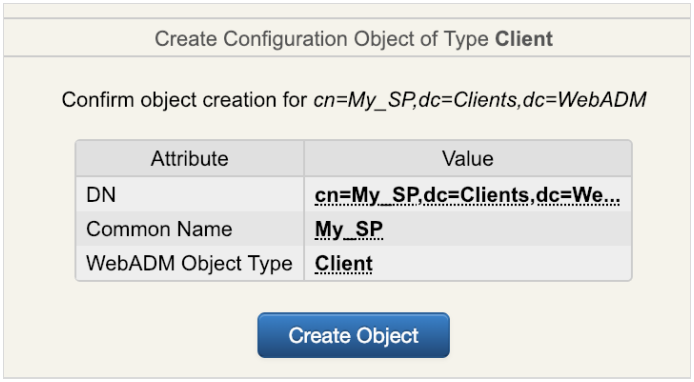
## 4. How to create a Client Policy per Service Provider (SP)

Since the WebADM 1.6.9-x and OpenID/SAML provider 1.3.0, it is possible to create WebADM client policies per Service Provider. That will allow you to return attributes, nameID, attributes mappings, or use a different certificate per client (SP) and not only globally. This feature makes the IDP much more powerful.

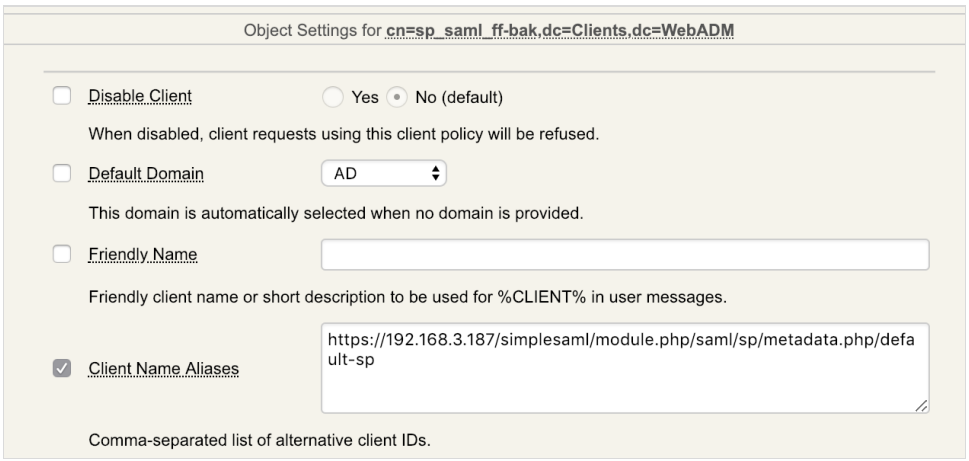
To create a client policy for your SP, log in on the WebADM Admin GUI, click on `Admin` tab, `Client Policy` and click on

`Add Client`.

Give a name to your `Client Policy` and then click `Proceed` and `Create Object`.

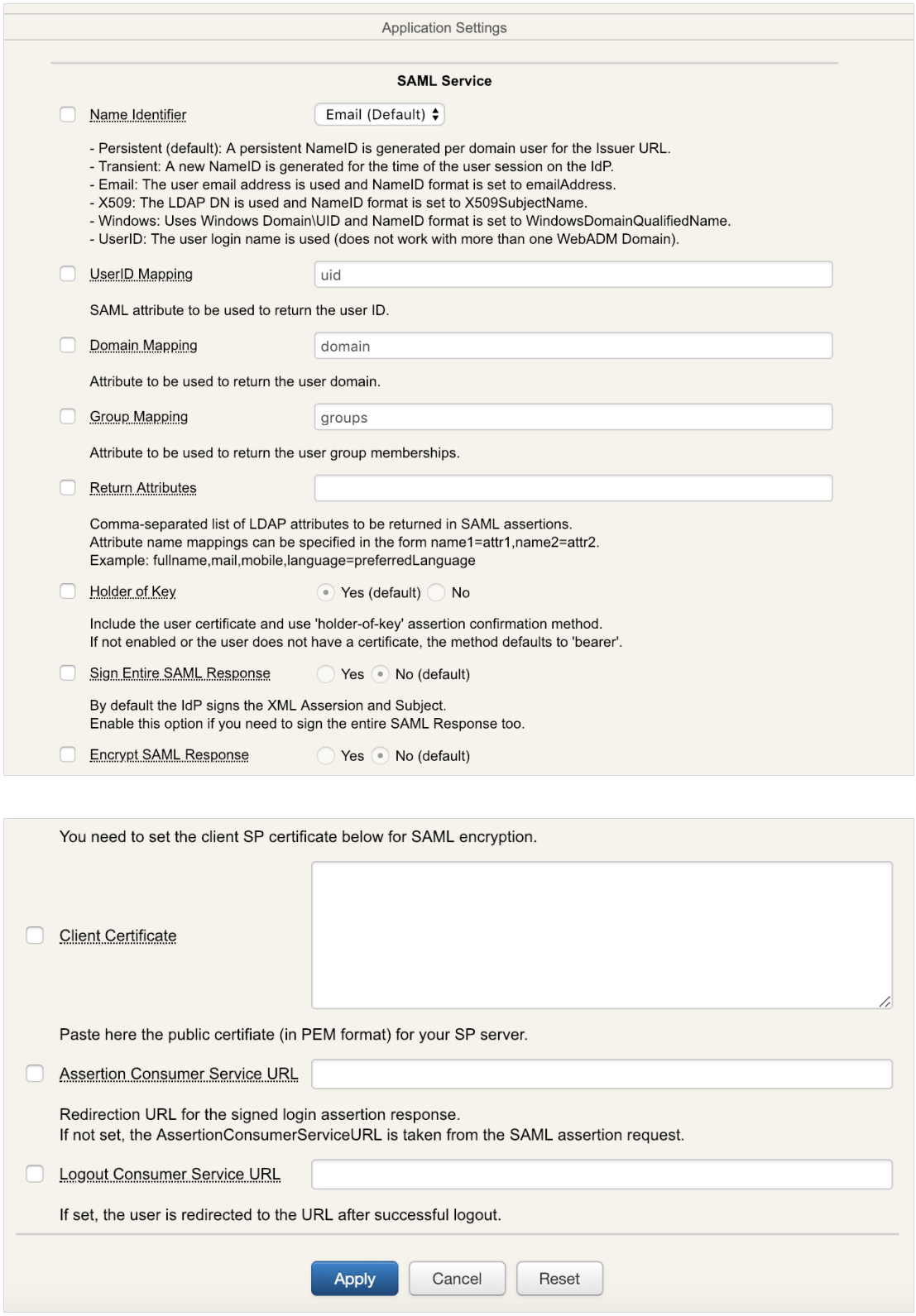



We will now configure the client policy. Many settings can be applied here like which users/groups/networks the client policy will be applied, allowed/excluded hours, which domain… An important setting on this page is the `Client Name Aliases` which will allow us to do the matching between the client policy and the SP. For this, the client policy must be created with the SP issuer URL (Entity ID) as `Client Name Aliases`.




The matching is done, we will now configure the SP policy.

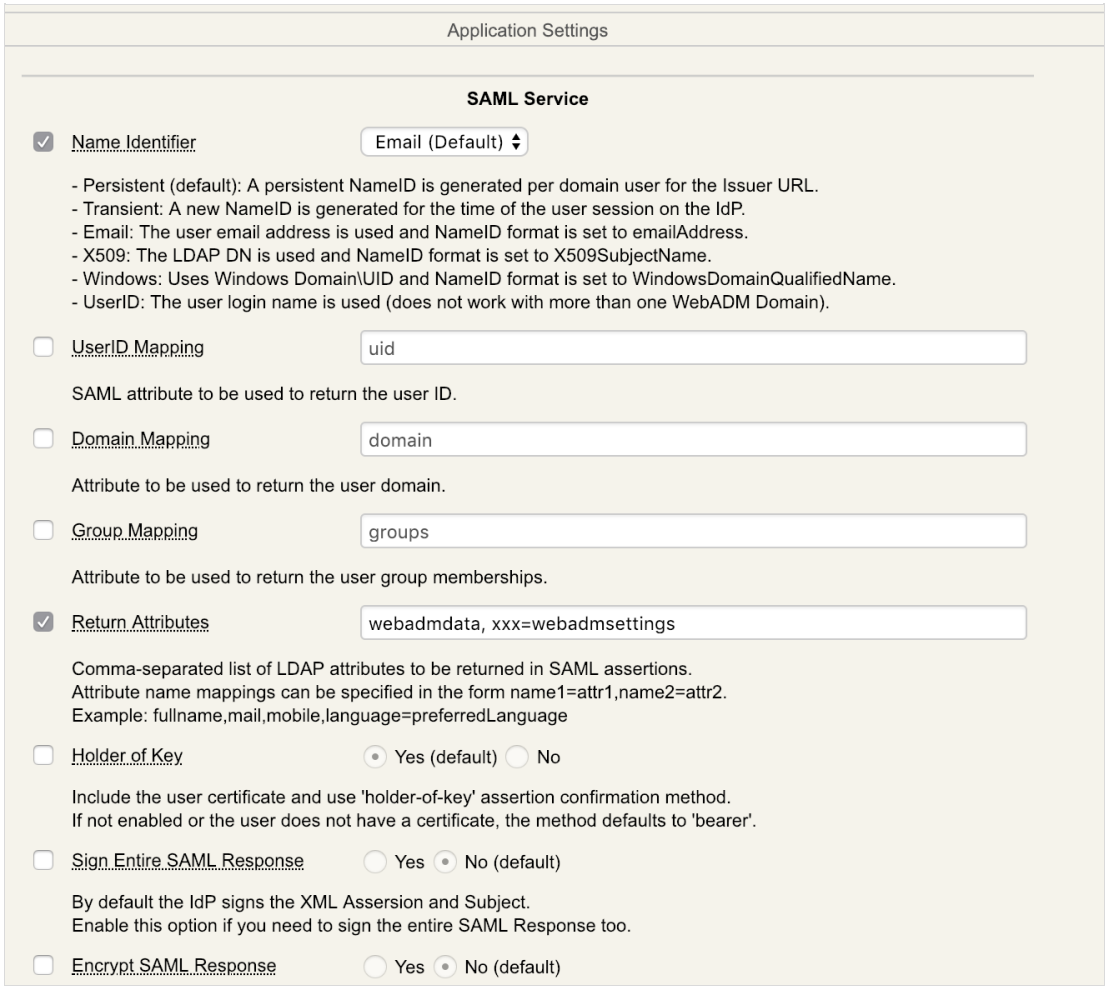
If you scroll down a little bit, you will find the setting named `Forced Application Policies`, click on the `Edit` button and select `OpenID` application in the left box.

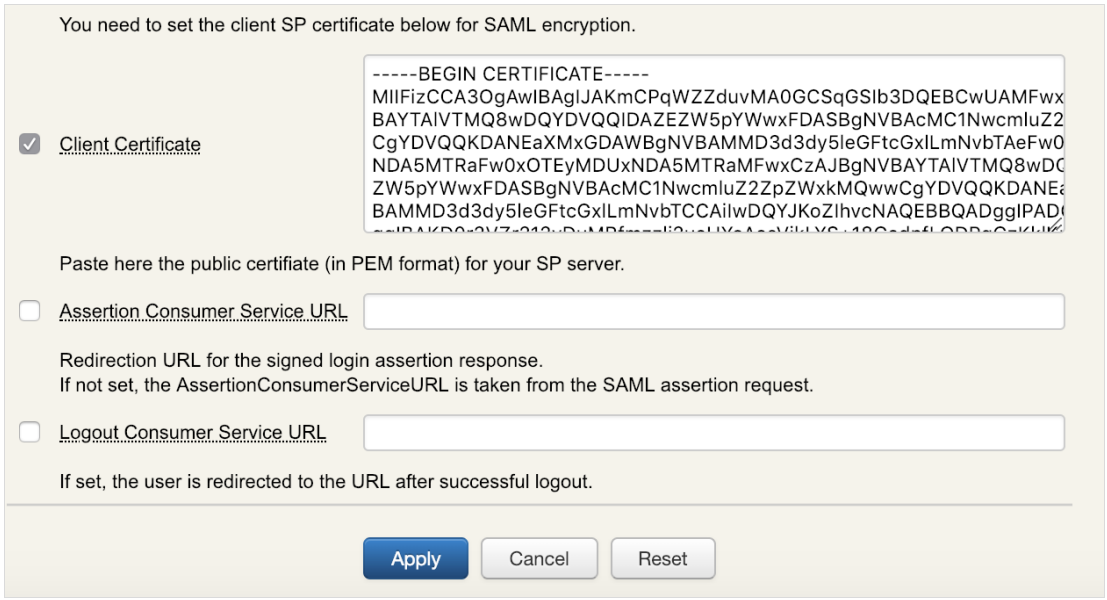






Configure your client policy with every setting you need for your SP and then save your configuration.




Your client policy for your SP is now configured. Try an authentication from your SP and check the WebADM logs to be sure that your policy is applied correctly.

## Note

You can not yet apply any OpenOTP settings in the same OpenID/SAML client policy. That part is in the RCDevs roadmap and will be added in the future.